<交大交交APP>

软件需求规约

版本 <2.1>

修订历史记录

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **版本** | **说明** | **作者** |
| 2019.6.1. | <1.0> | 初稿 | 林江浩 |
| 2019.6.12 | <2.0> | 立项答辩前的终稿 | 林江浩 |
| 2019.6.13 | <2.1> | 立项答辩后的修改稿，添加UI设计 | 林江浩 |
|  |  |  |  |

目录

[1. 简介 5](#_Toc11312775)

[1.1 目的 5](#_Toc11312776)

[1.2 背景 5](#_Toc11312777)

[1.3 定义、首字母缩写词和缩略语 6](#_Toc11312778)

[1.4 参考资料 6](#_Toc11312779)

[1.5 概述 6](#_Toc11312780)

[2. 整体说明 7](#_Toc11312781)

[3. 具体需求 8](#_Toc11312782)

[3.1 功能 8](#_Toc11312783)

[**3.1.1** <Use case 图> 8](#_Toc11312784)

[**3.1.2** <用户登录用例规约> 9](#_Toc11312785)

[**3.1.3** <用户申请预约用例规约> 9](#_Toc11312786)

[**3.1.4** <用户接受预约用例规约> 10](#_Toc11312787)

[**3.1.5** <用户确认交易用例规约> 10](#_Toc11312788)

[**3.1.6** <用户查询交易历史用例规约> 11](#_Toc11312789)

[**3.1.7** <用户评价交易用例规约> 11](#_Toc11312790)

[**3.1.8** <用户查询信用评价用例规约> 12](#_Toc11312791)

[**3.1.9** <用户管理个人买卖信息用例规约> 12](#_Toc11312792)

[**3.1.10** <用户查询当前买卖信息用例规约> 14](#_Toc11312793)

[**3.1.11** <用户接收系统推送用例规约> 14](#_Toc11312794)

[**3.1.12** <用户即时聊天用例规约> 15](#_Toc11312795)

[**3.1.13** <用户管理个人信息用例规约> 15](#_Toc11312796)

[**3.1.14** <用户举报投诉用例规约> 16](#_Toc11312797)

[**3.1.15** <管理员创建活动用例规约> 16](#_Toc11312798)

[**3.1.16** <管理员管理用户信息用例规约> 16](#_Toc11312799)

[**3.1.17** <管理员维护站点信息用例规约> 17](#_Toc11312800)

[**3.1.18** <管理员查看统计数据用例规约> 17](#_Toc11312801)

[**3.1.19** <管理员管理买卖信息用例规约> 18](#_Toc11312802)

[3.2 易用性 19](#_Toc11312803)

[3.3 可靠性 19](#_Toc11312804)

[3.4 性能 19](#_Toc11312805)

[3.5 可支持性 19](#_Toc11312806)

[**3.5.1** **<Golang编码规范>** 19](#_Toc11312807)

[3.6 设计约束 23](#_Toc11312808)

[3.7 接口 23](#_Toc11312809)

[**3.7.1** **用户界面** 24](#_Toc11312810)

[**3.7.2** **硬件接口** 25](#_Toc11312811)

[**3.7.3** **软件接口** 26](#_Toc11312812)

[**3.7.4** **通信接口** 26](#_Toc11312813)

[3.8 适用的标准 26](#_Toc11312814)

软件需求规约 (简化版)

# 简介

## 目的

项目名为交大交交APP，是一款交大的二手交易信息平台，此文档的目的在于提供关于交大交交APP的需求功能的详述以及系统性能参数的说明。

## 背景

二手交易是指消费者将购买于企业或代理商、零售商后使用一段时间，或者未使用过的商品，作为一种新的商品再进行交易流通的行为。大学生这一群体，生活区域相近，消费心理相似，购买的需求商品具有极高的重合度，有着极高的二手交易的潜在需求。譬如在开学季，大学生会寻求以低价购入相应的教材，在一个学期的使用后重新以二手书的形式重新售出；在毕业季，毕业生的大部分书籍、生活用品、自行车都可以以二手交易的形式售出，否则也只能当做废品丢弃或回收。

而目前，交大师生进行二手交易的主要渠道主要有三：二手交易群、个人在朋友圈或空间发布交易信息、以闲鱼为代表的传统二手交易APP。

以“上海交大二手交易群”为例，此交易群的群成员已经超过3000人（其中不乏教职工的存在），并且正在建立更多的分群，大群平均每日发布的二手交易信息接近百条，若是碰上开学季、毕业季等高峰时期，每日的交易信息可以接近千条。可见交大当下对二手交易的需求极大，且还有不断增长的趋势。

但随着二手交易群的不断推广，以及二手交易行为的频率不断上升，以上三种交易渠道的弊端也逐渐凸显，比如，二手交易群没有历史检索、商品分类、智能搜索等功能，朋友圈发布消息传播性极低，而以闲鱼为代表的传统二手交易APP又具有非本土化和无法发布求购信息的不足。

综上，鉴于二手交易在交大师生群体中存在极大的需求空间，以及考虑到当下以二手交易群为代表的二手交易途径的种种弊端，通过现代化技术手段优化传统群发盲寻的二手群交易模式，，开发一款实用、易用的二手交易信息平台APP，克服二手交易群现有的使用问题，以满足交大师生对二手交易更便捷体验的需求，是十分有必要的。

## 定义、首字母缩写词和缩略语

暂无

## 参考资料

暂无

## 概述

该文档主要分为三个部分，第一部分即简介，主要对该文档的目的及项目的背景进行简要介绍；第二部分为整体说明，从整体的视角描述项目的效果、功能、特征、约束等；第三部分为具体需求，详细描述了项目的功能需求、质量属性、设计约束等。

# 整体说明

交大交交APP产品以达成上海交通大学内与二手交易相关的最好的网站/APP为目标，支持卖家发布出售信息、卖家发布求购信息、智能检索、自动匹配买卖双方等功能，后续**可能**会朝校园租赁服务等方向扩展功能。

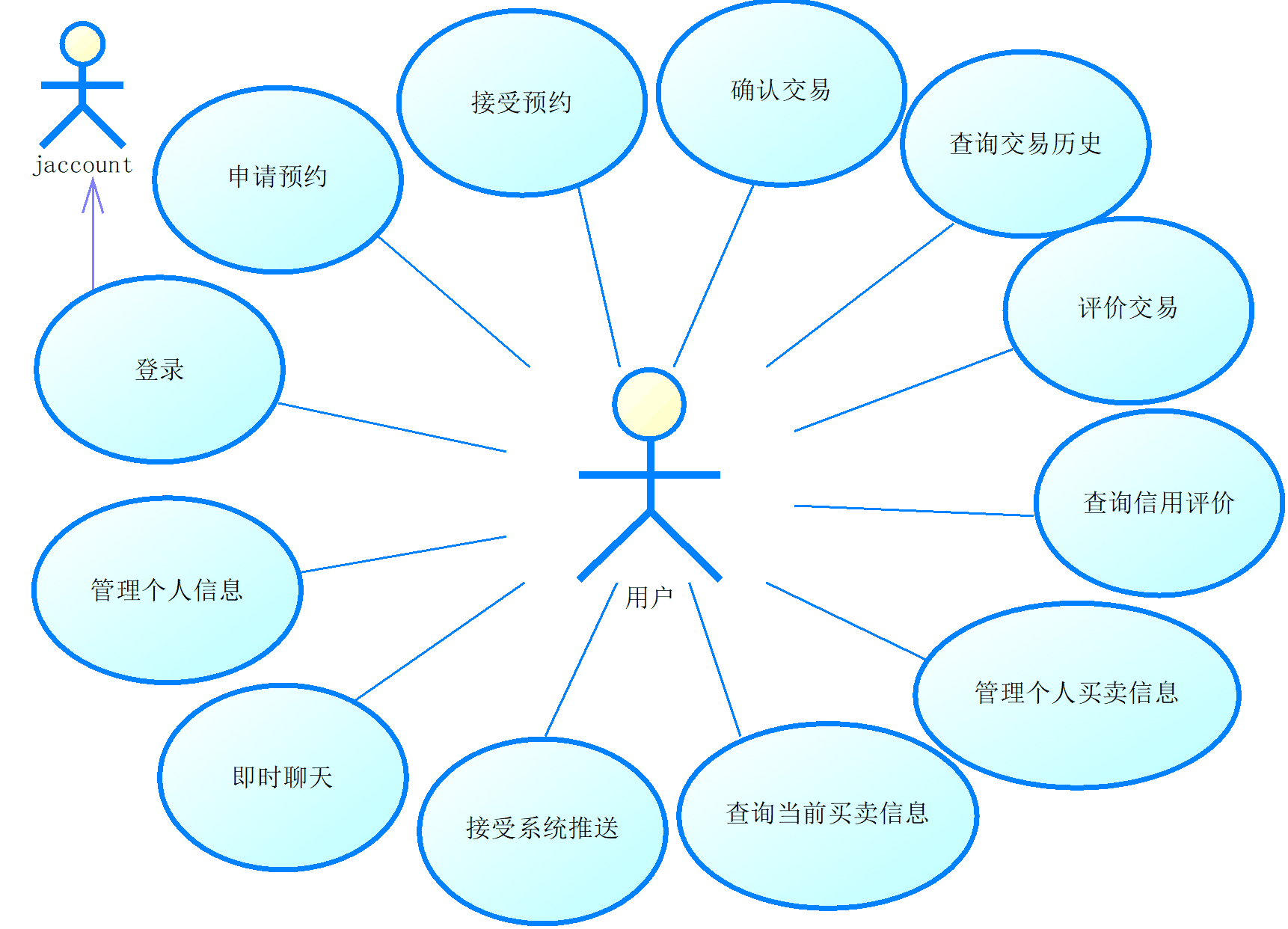
交大交交APP针对的用户群体广泛且稳定，为上海交通大学的在校师生，主要分布于闵行、徐汇两大校区，它的开发紧紧依赖于学生对更好的二手交易体验的需求，并以SJTU项目组的技术能力和开发经验为约束瓶颈。

交大交交APP开发时间预计为2个月，采用敏捷开发的Scrum软件过程，项目组为4人。

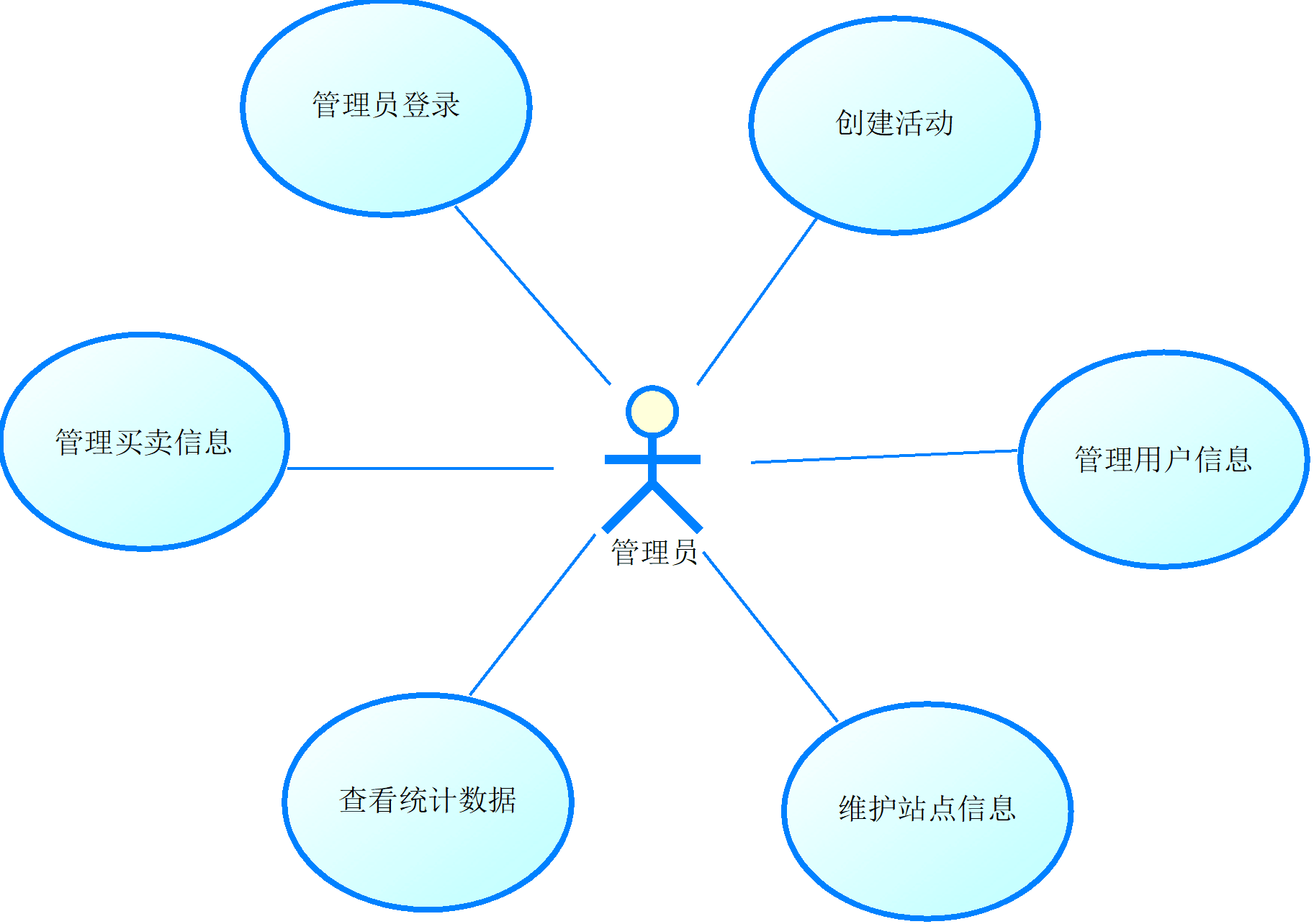
# 具体需求

## 功能

### <Use case 图>



**3.1.1.1 用户用例图**



**3.1.1.1 管理员用例图**

### <用户登录用例规约>

**概述：**所有用户都可以登录交大蕉蕉/娇娇APP

**执行者：**所有用户

**前置条件：**用户拥有JAcount账号

**后置条件：**无

**基本流：**

1. 系统提示用户选择通过第三方（JAccount）登录
2. 系统获取JAcount登录结果，提示用户下次是否自动登录
3. 用户选择自动登录，记录用户登录信息

**备选流：**

2.1 登录失败，提示用户登录失败

3.1 用户取消自动登录，不记录用户登录信息

### <用户申请预约用例规约>

**概述：**所有用户可以申请预约交易

**执行者：**用户

**前置条件：**用户已登录，买卖信息已存在，买卖信息处于可预约状态

**后置条件：**无

**基本流：**

1. 用户进入买卖信息查询详情界面
2. 用户输入己方交易信息
3. 用户点击申请预约按钮
4. 系统提示申请预约成功

**备选流：**

3.1 买卖信息不处于可预约状态 -> 系统提示“该交易不可预约”，用例结束

3.2 买卖信息不存在-> 系统提示“该交易已被撤销”，返回查询买卖信息用例

3.3 交易信息不符合格式-> 系统提示“交易信息格式错误”，返回到2

4.1 申请预约失败-> 系统提示“申请交易失败”，返回到3

### <用户接受预约用例规约>

**概述：**发布买卖信息的用户可以接受交易预约

**执行者：**用户

**前置条件：**用户已登录，买卖信息已存在，买卖信息处于可预约状态

**后置条件：**买卖信息进入已预约状态

**基本流：**

1. 用户进入买卖信息预约详情界面
2. 系统渲染收到的预约申请条目
3. 用户点击接受条目的预约按钮
4. 系统提示接受预约成功

**备选流:**

2.1 系统渲染失败->系统提示“查询接受的预约申请失败”，用例结束

3.1 用户点击拒绝预约按钮 -> 系统提示“拒绝预约”，预约信息被拒绝，返回到2

3.2 买卖信息已处于可预约状态 -> 系统提示“已预约”

4.1 接受预约请求失败-> 系统提示“接受预约失败”，返回到3

### <用户确认交易用例规约>

**概述：**已完成交易预约的用户可以确认交易完成

**执行者：**用户

**前置条件：**用户已登录，买卖信息处于已预约状态，

**后置条件：**买卖信息进入交易完成状态

**基本流：**

1. 双方用户进入已预约交易信息界面
2. 双方用户在长度小于阈值的时间段内同时点击结束交易按钮
3. 系统提示结束交易成功

**备选流:**

2.1 某方用户超时->系统提示“交易失败”，返回到2

3.1 提交结束请求失败->系统提示“提交请求失败”，返回到2

### <用户查询交易历史用例规约>

**概述：**用户可以查询自己的交易历史

**执行者：**用户

**前置条件：**用户已登录

**后置条件：**无

**基本流：**

1. 用户进入交易历史查询界面
2. 系统渲染交易历史条目

**备选流:**

2.1 系统渲染失败->系统提示“查询交易历史失败”，用例结束

4.1 提交评价失败->系统提示“提交评价失败”，返回到3

### <用户评价交易用例规约>

**概述：**已完成交易的用户可以评价交易

**执行者：**用户

**前置条件：**用户已登录，买卖信息处于交易完成状态，

**后置条件：**买卖信息进入已评价状态

**基本流：**

1. 用户进入已预约交易信息界面
2. 用户输入评价图片、文字与评价星级
3. 用户点击评价按钮
4. 系统提示评价成功

**备选流:**

2.1 上传评价图片失败->系统提示“请重新上传图片”，返回到2

2.2 上传图片超过大小-> 系统提示“图片超过大小限制”,返回到2

3.1 未输入评价星级->系统提示“请输入评分”，返回到2

3.2 买卖信息结束时间超过阈值->系统提示“交易结束过长时间”，返回到2

4.1 提交评价失败->系统提示“提交评价失败”，返回到3

### <用户查询信用评价用例规约>

**概述：**用户可以查询自己和他人的信用评价

**执行者：**用户

**前置条件：**用户已登录

**后置条件：**无

**基本流：**

1. 用户进入系统信息界面，选中某个用户
2. 系统渲染用户信息，并显示该用户的信用评价

**备选流:**

2.1 查询信用评价失败，系统提示“查询信用评价失败”

### <用户管理个人买卖信息用例规约>

**概述：**用户可以增删查改自己的买卖信息，根据其点击的按钮，如“创建”,进入用户创建买卖信息子流，“查询”，进入用户查询买卖信息子流，“删除”，进入用户创建删除信息子流。

#### <用户创建买卖信息>

**执行者：**用户

**前置条件：**用户已登录

**后置条件：**无

**基本流：**

1. 用户进入创建买卖信息界面
2. 用户输入买卖信息
3. 系统匹配存在的买卖信息
4. 用户点击创建按钮
5. 系统对所有匹配的买卖信息发起者发出推送

**备选流:**

3.1 用户点击匹配的信息->系统提示是否直接交易，用户如选择是，则进入申请预约用例，否则进入4

3.2 系统未匹配到信息->系统提示“无匹配交易信息”，进入4

4.1 创建请求失败->系统提示“创建未成功”，返回2

4.1 交易信息不符合格式->系统提示“交易信息格式错误”，返回2

#### <用户修改买卖信息>

**执行者：**用户

**前置条件：**用户已登录，买卖信息已存在

**后置条件：**无

**基本流：**

1. 用户进入查询买卖信息界面
2. 用户修改买卖信息
3. 系统匹配存在的买卖信息
4. 用户点击修改按钮
5. 系统对所有匹配的买卖信息发起者发出推送

**备选流:**

3.1 用户点击匹配的信息->系统提示是否直接交易，用户如选择是，则进入申请预约用例，否则进入4

3.2 系统未匹配到信息->系统提示“无匹配交易信息”，进入4

4.1 修改请求失败->系统提示“修改未成功”，返回2

4.2 交易信息不符合格式->系统提示“交易信息格式错误”，返回2

#### <用户查询买卖信息>

**执行者：**用户

**前置条件：**用户已登录，买卖信息已存在

**后置条件：**无

**基本流：**

1. 用户进入查询买卖信息界面

**备选流:**

1.1 用户点击查询预约按钮->进入用户接收预约用例

1.2 用户点击修改按钮->进入用户修改买卖信息用例

#### <用户删除买卖信息>

**执行者：**用户

**前置条件：**用户已登录，买卖信息已存在

**后置条件：**买卖信息进入被撤销状态

**基本流：**

1. 用户进入查询买卖信息界面
2. 用户点击删除按钮
3. 系统对所有预约申请者发出推送

**备选流:**

2.1 删除请求失败->系统提示“删除未成功”，返回1

### <用户查询当前买卖信息用例规约>

**概述：**用户进入页面，选择买家则进入查询卖家信息子流，选择卖家则进入查询买家信息子流

#### <用户查询买家信息>

**概述：**用户可以查询正在进行的买家信息

**执行者：**用户

**前置条件：**用户已登录

**后置条件：**无

**基本流：**

1. 用户进入买家界面
2. 系统向用户推荐可能需要的买家信息

**备选流:**

2.1 用户输入关键词，系统重新查询

2.2 用户选择筛选标签，系统重新查询

#### <用户查询卖家信息>

**概述：**用户可以查询正在进行的卖家信息

**执行者：**用户

**前置条件：**用户已登录

**后置条件：**无

**基本流：**

1. 用户进入卖家界面
2. 系统向用户推荐可能需要的卖家信息

**备选流:**

2.1 用户输入关键词，系统重新查询

2.2 用户选择筛选标签，系统重新查询

### <用户接收系统推送用例规约>

**概述：**用户可以查询被匹配的交易信息，并选择是否申请预约

**执行者：**用户

**前置条件：**用户已登录

**后置条件：**无

**基本流：**

1. 用户进入系统信息界面
2. 系统渲染交易信息推送条目

**备选流:**

2.1 用户点击选中的交易信息->进入申请预约用例

2.2 接收推送失败->系统提示“接收推送失败”，用例结束

### <用户即时聊天用例规约>

**概述：**用户之间可以进行简单即时聊天，若用户发起聊天，执行发起聊天子流；若用户拉取聊天内容，执行拉取聊天子流。

#### <发起聊天>

**概述：**用户向他人可以发起聊天

**执行者：**用户

**前置条件：**用户已登录

**基本流：**

1. 用户发送文字、图片
2. 系统接受消息，打上【未读】标签

**备选流：**

2.1 发起聊天流中若发起用户在接受用户黑名单中，系统将打上【拒绝】标签并向发起用户返回，退出流程。

#### <拉取聊天>

**概述：**用户可以拉取别人的聊天

**执行者：**用户

**前置条件：**用户已登录

**基本流：**

1. 用户接受他人的消息
2. 系统发送用户消息，打上【已读】标签

**备选流：**无

### <用户管理个人信息用例规约>

**概述：**用户可以管理其个人信息

**执行者：**用户

**前置条件：**用户已登录

**基本流：**

1. 用户点击修改个人信息
2. 用户修改信息
3. 系统验证信息合法性
4. 页面显示修改成功

**备选流：**

3.1 修改信息不合法，返回错误页面，用户选择修改则返回基本流2，放弃修改则退出用例

### <用户举报投诉用例规约>

**概述：**用户可以举报投诉其他用户

**执行者：**用户

**前置条件：**用户已登录

**基本流：**

1. 用户在商品界面选择举报投诉
2. 用户选择举报类型，上传证据，提交
3. 系统推送消息给管理员

**备选流：**

2.1 用户在举报后可以选择删除举报，退出流程。

### <管理员创建活动用例规约>

**概述：**管理员可以临时创建活动

**执行者：**管理员

**前置条件：**管理员已登录

**后置条件：**无

**基本流：**

1. 系统提示管理员输入活动名称、活动时间
2. 管理员输入，点击创建
3. 系统为app生成临时活动板块

**备选流:**

2.1 管理员未输入或输入不完整，系统提示“输入不完整”，返回1

2.2 管理员输入内容不合法，系统提示输入不合法，返回1

### <管理员管理用户信息用例规约>

**概述：**管理员可以查询修改用户信息，管理员进入管理用户信息界面后，选择“查询”进入查询信息子流，选择“修改”，对某个用户信息进行修改。

#### <管理员修改用户信息>

**执行者：**管理员

**前置条件：**管理员已登录

**后置条件：**无

**基本流：**

1. 管理员进入修改用户信息界面
2. 管理员修改用户信息
3. 管理员点击修改按钮
4. 系统更新用户信息

**备选流:**

4.1 修改请求失败->系统提示“修改未成功”，返回2

4.2 修改信息不符合格式->系统提示“用户信息格式错误”，返回2

#### <管理员查询用户信息>

**执行者：**管理员

**前置条件：**管理员已登录

**后置条件：**无

**基本流：**

1. 管理员选择筛选条件，点击查询
2. 系统返回查询结果

**备选流:**

2.1 查询无结果，系统提示“按此查询条件无结果”

### <管理员维护站点信息用例规约>

**概述：**管理员可以管理站点信息

**执行者：**管理员

**前置条件：**管理员已登录

**基本流：**

1. 管理员修改站点信息
2. 管理员点击保存
3. 页面显示修改成功

**备选流：**

1.1 管理员离开页面或者取消操作，退出流程

### <管理员查看统计数据用例规约>

**执行者：**管理员

**前置条件：**管理员已登录

**后置条件：**无

**基本流：**

1. 管理员进入统计信息页面
2. 系统返回统计数据，包括：用户数量及其交易数量统计，系统提示“正在处理，请耐心等待”

**备选流:**

2.1 响应超时，系统提示“响应超时”

### <管理员管理买卖信息用例规约>

**概述：**管理员可以查询修改删除买卖信息，管理员进入管理买卖信息界面后，选择“查询”进入查询信息子流，选择“修改”，对某个买卖信息进行修改，选择“删除”，删除某条买卖信息。

#### <管理员修改买卖信息>

**执行者：**管理员

**前置条件：**管理员已登录

**后置条件：**无

**基本流：**

1. 管理员进入修改买卖信息界面
2. 管理员修改买卖信息
3. 管理员点击修改按钮
4. 系统更新买卖信息

**备选流:**

4.1 修改请求失败->系统提示“修改未成功”，返回2

4.2 修改信息不符合格式->系统提示“买卖信息格式错误”，返回2

#### <管理员查询买卖信息>

**执行者：**管理员

**前置条件：**管理员已登录

**后置条件：**无

**基本流：**

1. 管理员选择筛选条件，点击查询
2. 系统返回查询结果

**备选流:**

2.1 查询无结果，系统提示“按此查询条件无结果”

#### <管理员删除买卖信息>

**执行者：**管理员

**前置条件：**管理员已登录，买卖信息已存在

**后置条件：**买卖信息进入被撤销状态

**基本流：**

1. 管理员进入查询买卖信息界面
2. 管理员点击删除按钮
3. 系统对所有相关人员发出推送

**备选流:**

2.1 删除请求失败->系统提示“删除未成功”，返回1

## 易用性

普通用户和高级用户高效地执行发布信息操作所需的培训时间不超过10分钟

用户界面的风格具有高度的一致性

## 可靠性

平均故障间隔时间（MTBF）为7天

平均修复时间（MTTR）为24小时

一年中交大交交APP正常运行时间至少达到98%

手机客户端的崩溃率不超过3%

每千行代码的错误数目为 4bug/KLOC

## 性能

交大交交APP对事务的平均响应时间为1s，最长响应时间为4s

交大交交APP每秒响应1000个用户请求

交大交交APP最大并发用户数为1000

## 可支持性

### **<Golang编码规范>**

**3.5.1.1.命名**

* 包

保持package的名字和目录保持一致，尽量采取有意义的包名，简短，有意义，尽量和标准库不要冲突。包名应该为**小写**单词，不要使用下划线或者混合大小写。

* 文件

尽量采取有意义的文件名，简短，有意义，应该为**小写**单词，使用**下划线**分隔各个单词。

* 结构体

采用驼峰命名法，首字母根据访问控制大写或者小写

声明和初始化格式采用多行

* 接口

声明和初始化格式采用多行

单个函数的结构名以 “er” 作为后缀，例如 Reader , Writer 。

* 常量

常量均需使用全部大写字母组成，并使用下划线分词，如果是枚举类型的常量，需要先创建相应类型

* 类

和结构体类似，变量名称一般遵循驼峰法，首字母根据访问控制原则大写或者小写，但遇到特有名词时，需要遵循以下规则：

如果变量为私有，且特有名词为首个单词，则使用小写，如 apiClient

其它情况都应当使用该名词原有的写法，如 APIClient、repoID、UserID

错误示例：UrlArray，应该写成 urlArray 或者 URLArray

若变量类型为 bool 类型，则名称应以 Has, Is, Can 或 Allow 开头

**3.5.1.3.注释（重要）**

* 包注释

每个包都应该有一个包注释，一个位于package子句之前的块注释或行注释。包如果有多个go文件，只需要出现在一个go文件中（一般是和包同名的文件）即可。 包注释应该包含下面基本信息(请严格按照这个顺序，简介，创建人，创建时间）：

包的基本简介（包名，简介）

创建者，格式： 创建人： rtx 名

创建时间，格式：创建时间： yyyyMMdd

*// util 包， 该包包含了项目共用的一些常量，封装了项目中一些共用函数。*

*// 创建人： xxx*

*// 创建时间： 20190601*函数注释

* 结构（接口）注释

每个自定义的结构体或者接口都应该有注释说明，该注释对结构进行简要介绍，放在结构体定义的前一行，格式为： 结构体名， 结构体说明。同时结构体内的每个成员变量都要有说明，该说明放在成员变量的后面（注意对齐）写在每个函数的头顶

* 函数（方法）注释

每个函数，或者方法（结构体或者接口下的函数称为方法）都应该有注释说明，函数的注释应该包括三个方面（严格按照此顺序撰写）：

简要说明，格式说明：以函数名开头，“，”分隔说明部分

参数列表：每行一个参数，参数名开头，“，”分隔说明部分

返回值： 每行一个返回值

* 代码逻辑注释

对于一些关键位置的代码逻辑，或者局部较为复杂的逻辑，需要有相应的逻辑说明，方便其他开发者阅读该段代码

* 注释风格

全部使用单行注释，禁止使用多行注释

中英文字符之间**严格使用空格分隔**

单行注释不要过长，禁止超过 120 字符

**3.5.1.3.代码风格**

* 语句

缩进直接使用 gofmt 工具格式化

折行方面，一行最长不超过120个字符

括号和空格方面，直接使用 gofmt 工具格式化，所有的运算符和操作数之间要留空格。

* 包

三种类型的包，标准库包，程序内部包，第三方包，每种之间用空行分隔

不要使用相对路径引入包

* 错误

不能丢弃任何有返回err的调用，不要使用 \_ 丢弃，必须全部处理。接收到错误，要么返回err，或者使用log记录下来

尽早return：一旦有错误发生，马上返回

用独立的错误流进行处理

* 测试

单元测试文件名命名规范为 example\_test.go

测试用例的函数名称必须以 Test 开头，例如：TestExample

重要的函数都要首先编写测试用例，测试用例和正规代码一起提交方便进行回归测试

## 设计约束

技术架构：微服务

建模工具：PowerDesigner、Axure、Processon

编程语言：Javascript、golang

前端框架：react native

后端框架：gin、go-micro、micro

数据库：mariadb、mongodb

版本控制：git（github）

自动化测试、CI/CD：Travis CI

IDE：VSCode、IntelliJ IDEA、Android studio

其他工具：docker、mergify、codecov

其他：Client端与Server端用HTTP和Restful Web Service

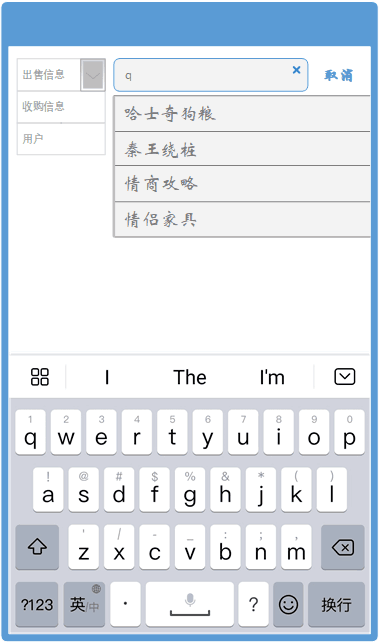
## 接口

暂无

### **用户界面**



**图3.7.1.1 首页界面设计图**

****

**图3.7.1.2 输入搜索界面设计图**

****

**图3.7.1.3 发布出售信息界面设计图**



**图3.7.1.4 即时聊天界面设计图**

### **硬件接口**

暂无

### **软件接口**

用户使用Android5.0或IOS10.0以上系统版本的手机客户端图形界面进行操作，管理员使用浏览器的图形界面进行后台管理

### **通信接口**

传输层协议为TCP

网络层协议为IP

前后端通信使用基于HTTP协议的Restful API

## 适用的标准

